

WebFOCUS

Defining a Word Substitution

EDA, FIDEL, FOCUS, FOCUS Fusion, Information Builders, the Information Builders logo, TableTalk, and Web390 are registered trademarks and SiteAnalyzer, SmartMart, WebFOCUS, and WorldMART are trademarks of Information Builders, Inc.

Acrobat and Adobe are registered trademarks of Adobe Systems Incorporated.

AvantGo is a trademark of AvantGo, Inc.

WebLogic is a registered trademark of BEA Systems, Inc.

HP MPE/iX is a registered trademark of Hewlett Packard Corporation.

Informix is a registered trademark of Informix Software, Inc.

Intel is a registered trademark of Intel Corporation.

ACF/VTAM, AIX, AS/400, CICS, DB2, DRDA, Distributed Relational Database Architecture, IBM, MQSeries, MVS, OS/2, OS/390, OS/400, RACF, RS/6000, S/390, VM/ESA, and VTAM are registered trademarks and DB2/2, Hiperspace, IMS, MVS/ESA, QMF, SQL/DS, VM/XA and WebSphere are trademarks of International Business Machines Corporation.

INTER SOLVE and Q+E are registered trademarks of INTER SOLVE.

Orbix is a registered trademark of Iona Technologies Inc.

ActiveX, FrontPage, Microsoft, MS-DOS, PowerPoint, Visual Basic, Visual C++, Visual FoxPro, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Motorola is a trademark of Motorola, Inc.

Teradata is a registered trademark of NCR International, Inc.

Netscape, Netscape FastTrack Server, and Netscape Navigator are registered trademarks of Netscape Communications Corporation.

Nextel is a registered trademark of Nextel Communications.

NetWare and Novell are registered trademarks of Novell, Inc.

Oracle is a registered trademark and Rdb is a trademark of Oracle Corporation.

Palm is a trademark and Palm OS is a registered trademark of Palm Inc.

PeopleSoft is a registered trademark of PeopleSoft, Inc.

INFOAccess is a trademark of Pioneer Systems, Inc.

BlackBerry is a trademark and RIM is a registered trademark of Research In Motion Limited.

SAP and SAP R/3 are registered trademarks and SAP Business Information Warehouse and SAP BW are trademarks of SAP AG.

Java and all Java-based marks, NetDynamics, Solaris, SunOS, and iPlanet are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Unicode is a trademark of Unicode, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2001, by Information Builders, Inc. All rights reserved. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Printed in the U.S.A.

Contents

Defining a Word Substitution1

 The Let Tool2

 The LET Command4

 Variable Substitution7

 Null Substitution9

 Multiple-line Substitution9

 Recursive Substitution 10

 Using a LET Substitution in a COMPUTE or DEFINE Command 11

 Checking Current LET Substitutions 12

 Interactive LET Query: LET ECHO 12

 Clearing LET Substitutions 13

 Assigning Phrases to Function Keys 14

Defining a Word Substitution

Topics:

- The Let Tool
- The LET Command
- Variable Substitution
- Null Substitution
- Multiple-line Substitution
- Recursive Substitution
- Using a LET Substitution in a COMPUTE or DEFINE Command
- Checking Current LET Substitutions
- Interactive LET Query: LET ECHO
- Clearing LET Substitutions
- Assigning Phrases to Function Keys

A LET substitution enables you to define a word to represent other words and phrases. By substituting words for phrases, you can reduce the typing necessary to enter requests (especially when entering phrases repeatedly) and make your requests easier to understand.

A substitution applies to a single procedure. You make a substitution using the LET command, the Let tool, or by placing an assignment in a file that you create with the text editor. You can run these assignments whenever you wish to apply them. You can also include the assignments (or run the file that contains them) in your profile; they will be applied automatically every time you start the product. For information about profiles, see your documentation on Dialogue Manager.

The Let Tool

The Let tool provides a graphical option for making a substitution.

Procedure How to Define a Substitution in the Let Tool

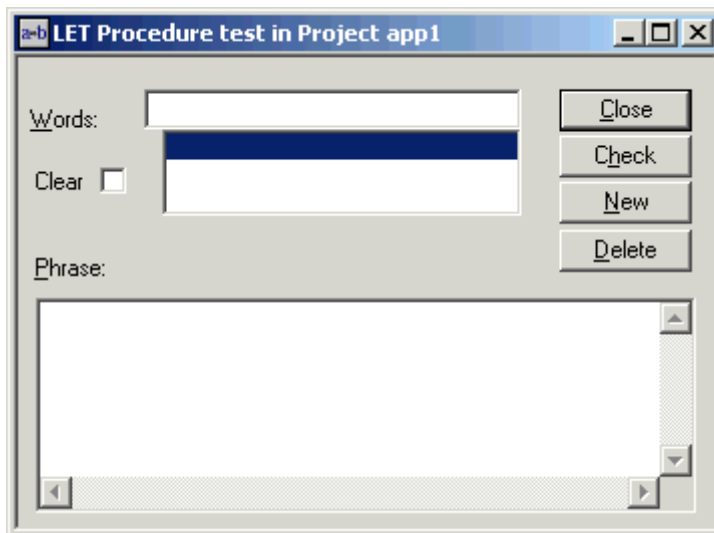
1. Right-click the procedure in the Procedures folder and choose *Open* from the shortcut menu. The Procedure window opens.
2. Click and hold a component connector (yellow diamond), then click the *Let* button on the Component Connector toolbox.



The Let tool opens. For details on the Let tool, see *Let Dialog Box* on page 1-2

3. Define your string substitution and click *Close*.
The Let dialog box closes.
4. When prompted to update the procedure, click *Yes*.

Reference Let Dialog Box



The Let dialog box has the following fields/options:

Words

Shows the current word for which you have defined a substitution. Below the current word is a list of other words for which you have defined substitutions. You can also enter a new word and a corresponding substitution phrase in the Phrase box.

Enter a word in the Words box that you wish to use in place of a longer phrase. Then enter the phrase in the Phrase box that the word will represent.

Clear

Clears a substitution that is already in effect.

Phrase

Displays the phrase that the word represents.

Close

Closes the Let tool. A dialog box prompts you to save your changes.

Check

Displays the LET command in code, and specifies any errors.

New

Activates the Words box so that you can add a substitution.

Delete

Removes a LET word and substitution phrase.

Procedure **How to Add a Let Substitution**

1. Click *New*.
The cursor appears in the Words box.
2. Enter a word that you wish to use in place of a longer phrase.
3. Use the Tab key or mouse to move the cursor to the Phrase box and enter the phrase that the word will represent.

Procedure **How to Edit a Word in a Let Substitution**

1. Select the word from the Words list.
The word appears in the Words box. The corresponding phrase appears in the Phrase box.
2. Use the Tab key or the mouse to move the cursor to the Words box.
3. Edit the word as needed.

Procedure How to Edit a Phrase in a Let Substitution

1. Select the word from the Words list.
The word appears in the Words box. The corresponding phrase appears in the Phrase box.
2. Use the Tab key or the mouse to move the cursor to the Phrase box.
3. Edit the phrase as needed.

The LET Command

The LET command enables you to represent a word or phrase with another word. This reduces the amount of typing necessary for issuing requests, and makes your requests easier to understand. A substitution is especially useful when you use the same phrase repeatedly. Note that you cannot use LET substitutions in Dialogue Manager commands, and substitutions cannot be used in a MODIFY request.

The LET command has a short form and a long form. Use the short form for one or two LET definitions that fit on one line. Otherwise, use the long form.

When you define a word with LET and then use that word in a request, the word is translated into the word or phrase it represents. The result is the same as if you entered the original word or phrase directly. You can substitute any phrase that you enter online unless you are entering a MODIFY request.

A LET substitution lasts until it is cleared or until the request terminates. To clear active LET substitutions, issue the LET CLEAR command. To use the same substitutions in many requests, place the LET commands in a stored procedure. If you want to save currently active LET substitutions, use the LET SAVE facility. These substitutions can then be executed later with one short command.

Syntax How to Make a Substitution (Short Form)

```
LET word = phrase [ ; word = phrase . . . ]
```

where:

word

Is a string of up to 80 characters with no embedded blanks.

phrase

Is a string of up to 256 characters, which can include embedded blanks. The phrase can also include other special characters, but semicolons and pound signs need special consideration. If the word you are defining appears in the phrase you are replacing, you must enclose it in single quotation marks.

More than one substitution can be defined on the same line by placing a semicolon between definitions.

Example Making a Substitution (Short Form)

The LET command defines the word WORKREPORT as a substitute for the phrase TABLE FILE EMPLOYEE:

```
LET WORKREPORT = TABLE FILE EMPLOYEE
```

Issuing the following

```
WORKREPORT
PRINT LAST_NAME
END
```

results in this request:

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
END
```

The next command includes TABLE as both the word you are defining and as part of the phrase it is replacing. It is enclosed in single quotation marks in the phrase:

```
LET TABLE = 'TABLE' FILE EMPLOYEE
```

More than one word is defined in the following command. The definitions are separated by a semicolon:

```
LET WORKREPORT=TABLE FILE EMPLOYEE; PR=PRINT
```

Syntax How to Make a Substitution (Long Form)

```
LET
word = phrase
.
.
.
END
```

where:

word

Is a string of up to 80 characters with no embedded blanks.

phrase

Is a string of up to 256 characters which can include embedded blanks.

END

Is required to terminate the command.

As shown, LET and END must each be on a separate line.

As with the short form, you can define several words on one line by separating the definitions with a semicolon. A semicolon is not required after the last definition on a line.

Example Making a Single Substitution (Long Form)

The following example illustrates a single substitution.

```
LET  
RIGHTNAME = 'STEVENS' OR 'SMITH' OR 'JONES' OR 'BANNING' OR 'MCCOY' OR  
'MCKNIGHT'  
END
```

Example Making Multiple Substitutions (Long Form)

The following example illustrates substitutions that span more than one line. Notice that there is no semicolon after the definition PR = PRINT:

```
LET  
WORKREPORT=TABLE FILE EMPLOYEE; PR = PRINT  
RIGHTNAME='STEVENS' OR 'SMITH' OR 'JONES'  
END
```

Example Defining Substitutions for Translation

Non-English speakers can use LET commands to translate a request into another language. For example, this request

```
TABLE FILE CAR  
SUM AVE.RCOST OVER AVE.DCOST  
BY CAR ACROSS COUNTRY  
END
```

can be translated into French as:

```
CHARGER FICHIER CAR  
SOMMER AVE.RCOST SUR AVE.DCOST  
PAR CAR TRAVERS COUNTRY  
FIN
```

Variable Substitution

Using the LET command, you can define a word that represents a variable phrase. A variable phrase contains placeholder symbols (carets) to indicate missing elements in the phrase. This allows you to give a phrase different meanings in different requests. Placeholders can be parts of words within phrases. They can also be used to represent system commands.

Placeholders can be unnumbered or numbered. If the placeholders are not numbered, then they are filled from left to right: the first word in the request after the LET-defined word fills the first placeholder, the second word fills the second placeholder, and so on to the last placeholder. If they are numbered, the placeholders are filled in numerical order. If you do not supply enough words to fill all the placeholders, the extra placeholders are null.

Example Making a Variable Substitution

The command

```
LET UNDERSCORE = ON < > UNDER-LINE
```

contains one placeholder. After issuing this command, you can use the word UNDERSCORE in a request:

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL BY EMP_ID BY HIRE_DATE
UNDERSCORE EMP_ID
END
```

The field name following the LET-defined word supplies the missing value to the placeholder. In the example, EMP_ID follows the defined word UNDERSCORE. This field name is inserted in the placeholder and translates UNDERSCORE EMP_ID as:

```
ON EMP_ID UNDER-LINE
```

Example Making Multiple Variable Substitutions (Unnumbered)

Issuing the LET command

```
LET TESTNAME = WHERE LAST_NAME IS < > OR < > OR < >
```

and then including the following line in a request

```
TESTNAME 'MCKNIGHT' 'STEVENS' 'BLACKWOOD'
```

translates the line as:

```
WHERE LAST_NAME IS 'MCKNIGHT' OR 'STEVENS' OR 'BLACKWOOD'
```

Notice that the variable phrase needs no placeholder at the end, and could also be code as WHERE LAST_NAME IS <> OR <>. Once all the placeholders are filled, the rest of the definition follows. In this example, the words MCKNIGHT and STEVENS would fill the two placeholders. BLACKWOOD would be left over, so it would follow the variable phrase.

If you do not supply enough words to fill in all the placeholders, the extra placeholders are null. For example, issuing this LET command

```
LET TESTNAME = WHERE LAST_NAME IS < > OR < > OR
```

and then entering this command

```
TESTNAME 'MCCOY'
```

translates the statement into:

```
WHERE LAST_NAME IS 'MCCOY' OR OR
```

This statement is illegal and produces an error message.

Example Making Multiple Variable Substitutions (Numbered)

The following LET command contains numbered placeholders:

```
LET TESTNAME = WHERE LAST_NAME IS <1> OR <2> OR <3>
```

Therefore, the following line

```
TESTNAME 'STEVENS' 'MCKNIGHT' 'BLACKWOOD'
```

is translated as follows:

```
WHERE LAST_NAME IS 'STEVENS' OR 'MCKNIGHT' OR 'BLACKWOOD'
```

If two placeholders have the same number, both placeholders are filled with the same word. For example, if you issue this LET command

```
LET RANGE = SUM MAX.<1> AND MIN.<1>
```

and this line

```
RANGE SALARY
```

the translated statement is:

```
SUM MAX.SALARY AND MIN.SALARY
```

Example Making a Variable Substitution in a Phrase

Issuing the following LET command

```
LET BIGGEST = MAX.< >
```

and entering the line

```
WRITE BIGGEST SALARY
```

translates the statement as:

```
WRITE MAX.SALARY
```

Example Defining a System Command

Each of the following LET commands define a system command in MVS:

```
LET ALFOC = TSO ALLOC F(< >) DA(< >.FOCUS) SHR
LET LISTMEM = TSO LISTDS < > MEMBERS
```

Null Substitution

With a null substitution, you can use more than one word to represent a phrase. By using more than one word in a request instead of a single word, you can make the request more readable.

You can define a null word using LET. A null word is ignored by the application.

Syntax How to Define a Null Word

To define a null word, issue the command

```
LET word=;
```

Example Defining a Null Word

This LET command defines DISPLAY as a null word:

```
LET
DISPLAY=;
AVESAL = SUM AVE.SALARY BY DEPARTMENT
END
```

In the following request, the word DISPLAY is used in the code DISPLAY AVESAL, for readability, to make clear that the request prints the value represented by AVESAL:

```
TABLE FILE EMPLOYEE
DISPLAY AVESAL
WHERE DEPARTMENT IS 'PRODUCTION'
END
```

The word DISPLAY is ignored and the request is translated as:

```
TABLE FILE EMPLOYEE
SUM AVE.SALARY BY DEPARTMENT
WHERE DEPARTMENT IS 'PRODUCTION'
END
```

Multiple-line Substitution

Many commands, such as END, must appear on a separate line in a report request. To include such a command in a LET definition, place a number sign (#) and a space before the command to indicate a new line. This allows you to substitute one word for several lines of code.

Example **Making Multiple-line Substitutions**

This LET command uses the number sign and a space to indicate that a new line is required for the END command:

```
LET HOLDREP = ON TABLE HOLD # END
```

The following request

```
TABLE FILE EMPLOYEE  
SUM AVE.GROSS BY EMP_ID BY PAY_DATE  
HOLDREP
```

is translated as:

```
TABLE FILE EMPLOYEE  
SUM AVE.GROSS BY EMP_ID BY PAY_DATE  
ON TABLE HOLD  
END
```

Recursive Substitution

Recursive substitution allows a phrase in one LET definition to contain a word defined in another LET definition. Recursive substitution can also be used to abbreviate long phrases within LET commands.

Example **Making a Recursive Substitution**

In the following LET command

```
LET  
TESTNAME=IF LAST_NAME IS RIGHTNAME  
RIGHTNAME = STEVENS OR MCKNIGHT OR MCCOY  
END
```

the word RIGHTNAME in the phrase in the first definition is defined in the second definition. (Note that the two phrases in the LET command could be reversed.) This LET command is equivalent to:

```
LET  
TESTNAME = IF LAST_NAME IS STEVENS OR MCKNIGHT OR MCCOY  
END
```

Example **Abbreviating a Long Phrase**

Consider the following LET command, which illustrates recursive substitution:

```
LET
TESTNAME = STEVENS OR SMITH OR MCCOY OR CONT1
CONT1    = BANNING OR IRVING OR ROMANS OR CONT2
CONT2    = JONES OR BLACKWOOD
END
```

You can use TESTNAME in this request:

```
TABLE FILE EMPLOYEE
PRINT SALARY BY LAST_NAME
IF LAST_NAME IS TESTNAME
END
```

This is the equivalent of:

```
TABLE FILE EMPLOYEE
PRINT SALARY BY LAST_NAME
IF LAST_NAME IS STEVENS OR SMITH OR MCCOY OR
BANNING OR IRVING OR ROMANS
OR JONES OR BLACKWOOD
END
```

Using a LET Substitution in a COMPUTE or DEFINE Command

A semicolon must follow an expression in a COMPUTE or DEFINE command. To use a LET substitution in a DEFINE or COMPUTE, you must include two semicolons in the LET syntax. You cannot create a LET substitution for a phrase that contains a semicolon.

Example Using a LET Substitution in a COMPUTE or DEFINE Command

The following LET syntax includes two semicolons, since the substitution will be made in a COMPUTE command:

```
LET
SALTEST = LEVEL/A4 = IF SALARY GT 35000 THEN HIGH
ELSE LOW;;
END
```

Issuing the command

```
AND COMPUTE SALTEST
```

translates the line into

```
AND COMPUTE LEVEL/A4 = IF SALARY GT 35000 THEN HIGH
ELSE LOW;
```

with one semicolon after the word LOW, as required by the expression in the COMPUTE.

Checking Current LET Substitutions

The ? LET command displays the currently active LET substitutions.

Syntax **How to Check Current LET Substitutions**

```
? LET [word1 word2 ... wordn]
```

where:

word1 word 2...wordn

Are the LET-defined words you want to check. If you omit these parameters, ? LET displays a two-column list of all active LET substitutions. The left column contains the LET-defined words; the right column contains the phrases the words represent.

Example **Checking Selected LET Substitutions**

Issuing

```
? LET CHART TESTNAME RIGHTNAME
```

displays a two-column list of the LET substitutions for CHART, TESTNAME, and RIGHTNAME.

Example **Checking All Current LET Substitutions**

Issuing

```
? LET
```

displays a list of all current LET substitutions.

Interactive LET Query: LET ECHO

The LET ECHO facility shows how FOCUS interprets FOCUS statements. This facility is a diagnostic tool you can use when statements containing LET-defined words are not being interpreted the way you expect them to.

When the LET ECHO facility is activated, when you enter a FOCUS statement, LET ECHO displays the statement as interpreted by FOCUS.

Reference **Results of LET ECHO commands**

The following explains the results of a LET ECHO command:

- If you enter a statement containing no LET-defined words, LET ECHO displays the statement as you entered it.
- If you enter a statement containing LET-defined words, LET ECHO displays the statement with the substitutions made.

- If the statement contains variable substitutions, LET ECHO displays the substitutions with the placeholders filled in.
- If the statement contains multiple-line substitutions, LET ECHO displays the statement with the substitutions on multiple lines.
- If the statement contains null substitutions, LET ECHO displays the statement with the LET-defined words deleted.
- If the statement contains recursive substitutions, the substitutions appear as they are finally resolved.
- LET ECHO may be coded at the top of a FOCEXEC. END ECHO gets coded on the last line of the FOCEXEC.

Note: If you enter a statement containing a variable substitution, you must enter as many words after the LET-defined word as there are placeholders in the phrase; otherwise, LET ECHO will wait for additional input.

Syntax **How to Activate the LET ECHO Facility**

To activate the LET ECHO facility, issue the command:

`LET ECHO`

Syntax **How to Deactivate the LET ECHO Facility**

`ENDECHO`

Clearing LET Substitutions

Use the LET CLEAR command to clear LET substitutions.

Syntax **How to Clear LET Substitutions**

`LET CLEAR { * | word1 [word2 . . . wordn] }`

where:

`*`

Clears all substitutions.

`word1 . . . wordn`

Are the LET-defined words that you want to clear.

Procedure **How to Clear a Substitution With the Let Tool**

1. Select the word from the Words list whose substitution you want to clear.
The corresponding phrase displays in the Phrases Box.

2. Select the *Clear* checkbox.
3. Click *Run*.

Example Clearing LET Substitutions

Issuing the following command

```
LET CLEAR CHART TESTNAME RIGHTNAME
```

clears substitutions for CHART, TESTNAME, and RIGHTNAME. If there are no additional LET substitutions in effect, the following command would have the same effect:

```
LET CLEAR *
```

Assigning Phrases to Function Keys

You can assign a phrase to a function key. Then when you have a blank line and press a function key, that phrase appears as if you actually typed it. This process works only in situations where the LET facility is operative.

Syntax How to Assign a Phrase to a Function Key

```
LET !n = [.]phrase
```

where:

n

Is a function key number from 1 to 24.

.

Suppresses the echo of the phrase when you press the function key.

phrase

Is the phrase that the specified function key represents.

Example Assigning Phrases to Function Keys

The following assigns values to function keys:

```
LET !4 = EX DAILYRPT  
LET !6 = END  
LET !20 = IF RECORDLIMIT EQ 10  
LET !21 = .EX MYREPORT
```

Index

Symbols

? LET command 12

C

commands

 ? LET 12

 ENDECHO 13

 LET 1

 LET ECHO 12

E

ENDECHO command 13

F

function keys

 assigning phrases to 14

L

LET command 1, 4 to 6, 9, 11, 14

Let dialog box 2 to 3

LET ECHO command 12 to 13

LET substitutions 1

 debugging 12

 displaying 12

 Let tool 2

Let tool 2 to 4

R

requests

 translating 6

S

substitutions 1, 4 to 9

 assigning to function keys 14

 debugging 12

T

tools

 Let 2

W

word substitutions 1, 4 to 9

 assigning to function keys 14

 debugging 12

